Organized Anonymous Agents

Warnier, M.E.; Brazier, F.M.

published in
the Proceedings of The Third International Symposium on Information Assurance and Security (IAS'07)
2007

document version
Publisher's PDF, also known as Version of record

Organized Anonymous Agents*

Martijn Warnier Frances Brazier
Intelligent Interactive Distributed Systems
VU University, Amsterdam
The Netherlands
{warnier,frances}@cs.vu.nl

Abstract

Anonymity can be of great importance in distributed agent applications such as e-commerce & auctions. This paper proposes and analyzes a new approach for organized anonymity of agents based on the use of pseudonyms. A novel naming scheme is presented that can be used by agent platforms to provide automatic anonymity for all *agents on its platform, or, alternatively, to provide anonymity* on demand*. The paper also introduces a new technique, based on the use of handles, that can be fully integrated in an agent platform. Performance measures for an anonymity service implemented for the AgentScape platform provides some insight in the overhead involved.*

1. Introduction

Agent technology provides state-of-the-art solutions for distributed applications such as e-commerce, e-health, e-government and electronic auctions [14]. The sensitive nature of data in these domains makes privacy an important requirement. Anonymity in multi agent systems can be acquired by conventional methods such as the use of a mediator or another outside trusted third party. A mediator or trusted third party acts on behalf of an agent (and its owner) and relays messages without revealing an agent's identity. However, such methods require explicit effort on the part of an agent application developer. He/she not only needs to design his/her application to relay all communication that needs to be anonymous to a mediator, (s)he also needs to ensure that no information regarding an agent's identity is leaked in the process.

This paper proposes a new approach to anonymous agent-to-agent communication that guarantees anonymity, (1) if required, for all agents running on a platform without any additional effort by agent application developers, or (2) on demand. The one main assumption is that agents trust the middleware on which they run –the agent platform.

The link between an agent and its owner does not have to be anonymous: the middleware is trusted and can thus be trusted to keep this information confidential[1].

Anonymity in the real world is not an absolute notion, communication can be anonymous to one person or organisation, and not to another. Similarly, agents can communicate anonymously to other agents, or groups of agents, but not, for example, to the agent platform on which they run. Of the several degrees of anonymity which can be distinguished ranging from absolute anonymity to total non-repudiation [4, 9] this paper focuses on *organized pseudonym-based semi-anonymity*. Semi-anonymity is organized: when and where anonymity is provided is well-defined. The naming scheme this paper introduces ensures that each agent's true identity and its pseudonyms are unique and cannot be linked to each other by any outside party.

Our approach provides a dedicated and organized solution for anonymity of agents, in contrast to more general anonymizing techniques. Most notably, Korba, Song and Yee [11] use onion routing [6] for all inter-agent communication. Onion routing provides anonymous communication by redirecting messages via a number of routers using an unpredictable path. Implemented in the JADE agent platform [3] a dedicated communication layer facilitates all anonymous inter-agent communication. This onion-routing approach can guarantee the same level of anonymity as the approach proposed in this paper. The difference, however, is that our approach provides a dedicated solution for agents

*This paper extends preliminary work reported at EUMAS'06 [23]

[1] In certain classes of applications agent platforms cannot accept agents that are completely anonymous for a number of reasons. The most simple is related to malicious agents: if such agents run havoc on an agent platform and their owners cannot be traced, an agent platform owner is unable to take any (legal) action against the agent owner, nor can it recognize future malicious agents coming from the same source. The need for agent platforms to be able to trace agent owners, as well as other identity management facilities [5], are assumed in this paper.

that can be fine-tuned to meet a range of agent specific settings, ranging from automatic anonymous communication between all agents, to per message anonymity.

The next section provides a more detailed explanation of anonymity as commonly defined in Computer Science focusing on its relevance to distributed agent systems. A naming scheme for anonymity is introduced, together with the aforementioned technique through which communication anonymity in agent systems can be acquired. The range of options: from complete integration in agent platform middleware, to solutions based on individual agent implementations, are discussed. The actual implementation of one of these options, an anonymizer service, in the AgentScape platform, illustrates the performance overhead involved.

The papers ends with discussion, conclusions and proposals for future work.

2. Anonymity in agent systems

Classical anonymity in Computer Systems focuses on anonymity of the underlying communication layer [21], as does this paper. The typical goal is to anonymously browse the Internet, or communicate with other parties without revealing the parties true identity. The related notions of anonymity are:

- sender anonymity
- receiver anonymity
- link anonymity (unlinkability)

The first two, sender and receiver anonymity, require that the location of the sender and receiver, respectively, are hidden from the other communicating party. Link anonymity, also known as unlinkability, ensures that the link between the communicating parties remains anonymous to all third parties: it is impossible for any outside party to observe if two parties are communicating with each other. (Note that the communicating parties themselves are often aware of each other's (true) identity.) In practice these forms of anonymity can be combined.

This general notion of anonymity in Computer Science can be applied to agent technology. The focus in this paper is on anonymity of communication between individual agents. As stated earlier the relation between agent owner and agent is assumed to be confidential, and guaranteed by the agent platform. Full communication anonymity, i.e., receiver, sender and link anonymity taken together, can only be established when an agent cannot be linked to a legal entity via its communication with other agents. A platform based solution that enables the middleware to (automatically) provide link anonymity and location anonymity for each individual agent is the main focus of this paper. Pseudonyms are introduced for this purpose.

The use of a pseudonym, however, on its own does not suffice. If outsiders can observe agent communication, these observations can be used to obtain/deduct (unwanted) information about an agent. If an agent uses the same pseudonym to communicate with several other agents, together they can infer that they have been talking to the same party which breaks anonymity.

In agent systems that support mobility of agents yet another form of anonymity exists: migration anonymity. In essence this hides the migration path of an agent, and thus hides the original starting platform of an agent which results in a form of anonymity. Migration anonymity is, however, outside the scope of this paper, see Rafal Leszczyna and Janusz Górski [13] for more detail on migration anonymity.

3. A naming scheme for anonymity

An agent platform can identify an agent by its *globally unique identifier*(GUID). This GUID corresponds uniquely to the identity of the agent. The following example illustrates how the use of one single pseudonym for all communication as discussed –briefly– above, does not suffice in multi-party negotiation situations:

Example 1
There are three agents A, B and C, each with their own pseudonym P_A, P_B and P_C respectively. Agent A is interested in a service that both agent B or C can provide. Agent A first uses its pseudonym P_A to ask agent B about the price of its service, then agent A uses the same pseudonym P_A to ask agent C about the price of its services. Although agent B and agent C do not know A's real identity, together they can still determine that the same agent *has been asking price information of the services they provide. Thus agent A has not been communicating anonymously.*

Example 1 above clearly demonstrates the need for agents to use more than one pseudonym to obtain (link) anonymity - one pseudonym for each individual communication event (or communication session). For similar reasons, agents should also use a different pseudonym each time they communicate with the same party at some later point in time.

The example also illustrates that privacy protection against buyer profiling cannot be obtained by solely using *one* pseudonym. As the same pseudonym can be linked to multiple events over a longer period of time, a buyer profile can be constructed, and privacy cannot be guaranteed. Even if the 'real' identity of the agent owner is not known!

In our approach each agent has one globally unique identifier and multiple unique pseudonyms. The agent platform is responsible for ensuring that all GUID's and

pseudonyms are unique, that GUID's are hidden from other agents, that pseudonyms cannot be linked to each other and that pseudonyms cannot be linked to the agent they represent.

This naming scheme is implemented using handles. A handle is a unique and meaningless string [2] that is bound to a specific agent.

Each agent is assumed to have a global unique identifier (GUID) known only to the agent platform. Such a GUID can, for example, be implemented by a Universally Unique Identifier (UUID, ISO 11578:1996). Furthermore, each agent can acquire as many (globally unique) handles as it requires. These handles serve as pseudonyms and are used for communication purposes.

As handles have no intrinsic meaning and do not leak any information about an agent or its owner, agents can safely use handles as pseudonyms. Link anonymity is acquired if agents use a new handle for each individual communication event.

The agent platform is responsible for creation of agent GUID's and handles and the binding between the two. The binding between handles and GUID's can be acquired using a cryptographic hash function [10]. The following algorithm can be used by an agent platform to generate handles for agents, where 'sha' is a cryptographic hash function:

$\text{handle}_1 = \text{sha}(\text{GUID} + 1)$
$\text{handle}_2 = \text{sha}(\text{GUID} + 2)$
...
or more generally stated:

$\text{handle}_n = \text{sha}(\text{GUID} + \text{n})$ with $n \in \mathbb{N}^+$

This approach has two specific advantages:

- If the GUID is not known then handles cannot be linked to each other or one specific GUID.
- If the GUID is known then the platform cannot deny that a specific handle belongs to a specific GUID.

Several properties of cryptographic hashes, such as Sha-1 and MD5 are used: First the fact that cryptographic hashes are *one way* is used, i.e., easy to compute but very hard (computably infeasible) to reverse. This guarantees that if an agent knows a handle it cannot compute the corresponding GUID. Cryptographic hashes also have the property that they are *collision free*, i.e., it is computably infeasible to construct two different GUID's that have the same hash-image. This ensures that all handles are unique. And cryptographic hashes also have the property that a *small change* in the input results in a significant change in the output (roughly half of the bits should change). This ensures that agents cannot determine if two handles belong to the same GUID (agent). Because handles are uniquely coupled to the GUID of an agent is not possible for an agent to (ab)use a handle of another agents.

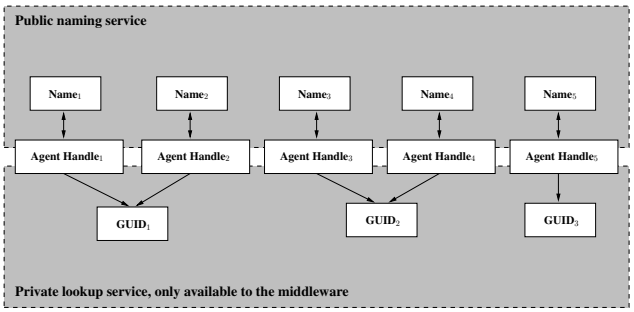

Figure 1. Schematic overview of the handle technique

An agent platform, however, given an agent's handle, must be able to retrieve its GUID. A private lookup service, as depicted in Figure 1 provides this functionality. This service must be private to the middleware. Note, that an agent platform can always check the integrity of its own lookup service should it doubt the information it acquires. An agent platforms can always reconstruct an agent's handles given its GUID as described above and compare it to the handle it has been provided.

Agents can also opt to use human readable names instead of the meaningless strings they are assigned as handles. Figure 1 depicts the mapping between human readable names and handles, and the *public* naming service in which the unique mappings between human readable names and handles, are stored and from which they can be retrieved.

Figure 1 schematically displays the handle technique at both levels. Both the public and the private lookup service are shown, as are the possible lookups depicted by the arrows.

As handles (or human readable names that are uniquely mapped to these handles) are used for all communication between agents, no information about the location of a particular agent is revealed to any other agent. Hence this technique also provides location anonymity and thus also sender and receiver anonymity. Whenever two agents communicate they do not have to share information on the location (host) on which they reside.

Agents can implicitly revoke a handle (simply by no longer using it) or explicitly (in which case the handle is removed from the private lookup service on the platform on which it runs). By definition, handles are also unique over time, due to the large number of possible handles, the same handle is never used twice. However, if so required, a time-stamping mechanism can be used to limit the lifetime of individual handles and hence make handles applicable for reuse. Note that guaranteeing uniqueness of handles is

important for reliability, correctness and accountability.

4. Middleware implementation

The technique described above can be implemented in several ways, some fully automatic, others on demand, and all but the most fine-tuned approaches as middleware services.

An organized solution for semi-anonymity requires namely a solution that is *integrated in the middleware* of the agent system. This provides the option for all communication between agents to be completely anonymous without any additional effort by the agent developer.

It is, however, often not necessary or desirable to make all communication anonymous. In fact, in many cases an agents may wish to reveal its identity. The reasons may vary considerably: to access information services for which intellectual property rights play a role, to negotiate a time slot with another agent, to collaborate with other agents to perform a specific task. An agent may also wish to establish several (virtual) identities for different situations. Such cases require a more fine-grained solution than fully automatic anonymization.

At the level of the middleware, *policies* can be defined that support different strategies for communication anonymization on a per agent basis. More advanced policies, e.g. supporting self-learning or self-organizing strategies for anonymization [24], can refine this further and allow agents to be anonymous to some agents while not anonymous to others.

A middleware *anonymizing service* –possibly implemented by an agent– is another option. A dedicated anonymizing service works as a sort of proxy that routes all communication it receives, using a new handle for each new communication session. This 'gateway' approach has the obvious advantage that agent developers have more fine tuned control of anonymity. The anonymizing service can be used when circumstances so require. It may also be the default option for all communication events, depending on the policies defined, thereby integrating the service complete into the middleware.

5. AgentScape

AgentScape[2] is a framework for development and deployment of open, large-scale distributed agent systems and includes support for fault-tolerance, security, heterogeneity and interoperability [16]. AgentScape's middleware security features include separate use of globally unique identifiers (GUID's) and handles (of agents and services), leasing of resources, sandboxing of agents, signing agent's code and its state, and secure communication. In AgentScape handles can be used for implementation of anonymity as middleware services without changing existing functionality.

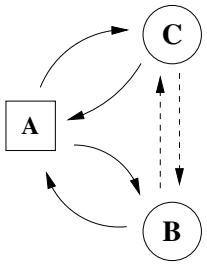

Figure 2. Anonymizing service A facilitates anonymous communication for agent B with agent C. The dashed arrows represent logical communication (between agent B and agent C) and the solid arrows represent the real message flow via the anonymizer (A).

AgentScape uses a handle-model as described above. Each agent has its own globally unique identifier (GUID). The GUID is generated upon creation of the agent and is kept private to the middleware. An agent always has an initial handle that can be linked —by the middleware– to an agent's GUID. Optionally, a name can be assigned to an agent's handle. Additional handles can be requested from the middleware. An agent's handles and names are registered in a Name Look-up Service (NLS), that is assumed to be private to the middleware[3].

An anonymizing service is provided by the AgentScape platform[4]. This service is implemented by means of a simple router, depicted in Figure 2 and illustrated in the following example:

Example 2
There is one anonymizing service[5] **A** *and two agents* **B** *and* **C** *in Figure 2 each with their own handles* H_A, H_B *and* H_C *respectively. Additional handles are numbered sequentially, thus* H_{An+1} *is the* $n_{th}+1$ *handle of anonymizer* **A**.

[2] http://www.agentscape.org

[3] Two different lookup services can be used in AgentScape: the *default public lookup services*, that is completely open and accessible and *a secure, decentralized lookup service* [22] that is only accessible to the AgentScape middleware.

[4] AgentScape version 0.9.0beta2 and upwards, available at http://www.agentscape.org, contains this anonymizing service.

[5] The anonymizing service in AgentScape is implemented by an agent. However as far as other agents are concerned this service is simply part of the platform middleware. Technically this is accomplished by removing (all) the handle(s) of the anonymizing service from publicly available *name services* and publishing the handle of the anonymizing service in a *yellow page service* [15].

Single Host; anonymizing disabled

# msgs	# threads	time					avg	dev	
1000	1	0.123	0.063	0.058	0.059	0.067	0.074	0.025	
500	2	0.065	0.065	0.066	0.064	0.065	0.065	0.001	
200	5	0.063	0.061	0.061	0.063	0.063	0.062	0.001	
100	10	0.060	0.061	0.068	0.062	0.066	0.063	0.003	

Single Host; anonymizing enabled

# msgs	# threads	time					avg	dev	overhead
1000	1	0.134	0.144	0.140	0.138	0.143	0.140	0.004	+89%
500	2	0.130	0.128	0.127	0.144	0.131	0.132	0.006	+103%
200	5	0.114	0.106	0.103	0.096	0.105	0.105	0.006	+62%
100	10	0.098	0.100	0.106	0.103	0.097	0.101	0.003	+60%

Sender and Receiver on separate hosts ; anonymizing disabled

# msgs	# threads	time					avg	dev	
100	1	15.53	15.55	15.81	15.93	15.73	15.71	0.15	
50	2	8.93	8.30	8.28	8.83	8.87	8.64	0.29	
20	5	7.24	7.37	7.35	7.32	7.43	7.34	0.06	
10	10	7.58	7.75	7.59	7.52	7.58	7.60	0.08	

Sender, Receiver and Anonymizer on separate hosts; anonymizing enabled

# msgs	# threads	time					avg	dev	overhead
100	1	24.71	24.86	24.65	24.70	24.79	24.74	0.07	+57%
50	2	15.97	13.92	15.35	14.98	15.97	15.24	0.76	+76%
20	5	15.83	16.16	16.31	16.14	16.03	16.10	0.16	+119%
10	10	15.24	14.66	15.89	16.94	16.26	15.80	0.79	+108%

Figure 3. Performance measurements of the anonymizer service depicted in Figure 2

Assume that agent **B** *wants to communicate with agent* **C** *using the anonymizer* **A**. *To this purpose, in its message (the* `Envelope`*), Agent* **B** *not only specifies, the 'to' (H_C) and 'from' (H_B) handles, but also a 'via' field for the anonymizer's handle (H_A). Anonymizer* **A**, *in turn, generates a new handle H_{An} for this message, stripping agent* **B***'s handle (H_B) from the message and forwarding the message to agent* **C** *using only this new handle (H_{An}). If Agent* **C** *chooses to reply to the message, agent* **C***'s reply is sent 'to' (H_{An}), 'from' (H_C) via anonymizer* **A**. *Anonymizer* **A** *forwards the reply to agent* **B**.

Agent **B** *can decide when the communication session ends. For each new session (with the same or another agent) a new handle is generated by the anonymizer* **A**.

The performance overhead of the anonymizer service is largely determined by the network: If either the anonymizer service and the sender (**A** and **B**), or the anonymizer service and the receiver (**A** and **C**), run on the same host, the performance overhead is reasonable (60%) as long as the number of messaging threads is large enough (1 thread per 10 concurrent messages). If both agents and the anonymizer service run on a different host there is a noticeable overhead (between 57% and +119%, roughly doubling the round-trip time of a single message send via the anonymizer), due to the two additional messages that are sent over the network. However, if each host runs its own middleware anonymizing service this overhead can be avoided. The performance, in this case, is only restricted by the resources (number of processes, memory or cpu time) of a particular host. Figure 3 shows the performed measurements and their results.

For similar reasons, the implementation scales well. As each host can run, at negligible cost, one or more instances of the middleware anonymizing service. In practice, this makes the network and machine resources the main bottleneck for scalability.

6. Discussion

This paper introduces a new anonymizing approach for agent systems that guarantees organized semi-anonymity for individual agents. Such anonymous agents can be used to ensure the privacy of users, e.g. with respect to service providers.

The proposed technique for acquiring anonymity, based on handles, can be completely integrated into agent platform middleware as a separate middleware service, as described for AgentScape. The maximum performance penalty for this service is a factor of two overhead for agent communication.

Although the approach discussed is theoretically secure, in practice a number of risks still remain. If the number of agents on an agent platform is known then, in theory, it is possible that all agents conspire together against one agent. This breaks link anonymity. A simple solution to this problem is to use a number of 'dummy' agents that belong to the agent platform itself. Other agents cannot determine if an agent is real or belongs to the platform an thus the attack no longer works.

Another risk, although highly unlikely due to its intrinsic properties, is the use of side channel attacks [12]. Timing attacks, as discussed in [17], form a particular challenging problem. Using a combination of techniques that observe timing behavior together with statistical analysis almost certainly breaks anonymity.

7. Future Work

Anonymous agents are particularly useful when deployed in open environments. However, in cases where agents need to authenticate themselves, e.g. to obtain access to a service, the anonymity of individual agents, and thus the privacy of its users, can no longer be upheld. Current research focuses on the use of *derived access tokens*, comparable to diversified keys [1], that can be used to gain access to a services without revealing ones 'true' identity. Similarly, (semi-)anonymous electronic payment systems [7] are needed if privacy and anonymity are required. Integrating such systems forms another challenge that is left for future research.

Acknowledgments

This research is conducted as part of the ACCESS project[6] funded by the NWO TOKEN program. The authors thank Stichting NLnet for their support, David de Groot and Benno Overeinder for their input and comments and

[6] `http://www.iids.org/access`

Reinier Timmer for the realization of the implementation in AgentScape.

References

[1] R. Anderson. The formal verification of a payment system. In M. G. Hinchey and J. P. Bowen, editors, *Industrial Strength Formal Methods*, pages 43–52. Springer-Verlag, Sept. 1999.

[2] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish. A layered naming architecture for the internet. *SIGCOMM Comput. Commun. Rev.*, 34(4):343–352, 2004.

[3] F. Bellifemine, A. Poggi, and G. Rimassa. JADE–A FIPA-compliant agent framework. *Proceedings of PAAM*, 99:97–108, 1999.

[4] F. Brazier, A. Oskamp, J. Prins, M. Schellekens, and N. Wijngaards. Anonymity and software agents: An interdisciplinary challenge. *AI & Law*, 1-2(12):137–157, 2004.

[5] D. de Groot and F. Brazier. Identity Management in Agent Systems. In N. Foukia, J. Seigneur, and M. Purvis, editors, *Proceedings of the First International Workshop on Privacy and Security in Agent-based Collaborative Environments (PSACE)*, pages 23–34, 2006.

[6] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*, volume 2, 2004.

[7] F. D. Garcia and J.-H. Hoepman. Off-line Karma: A Decentralized Currency for Peer-to-peer and Grid Applications. In J. Ioannidis, A. Keromytis, and M. Yung, editors, *3rd Applied Cryptography and Network Security (ACNS 2005)*, volume 3531 of *LNCS*, pages 364–377. Springer-Verlag, 2005.

[8] J. A. Goguen and J. Meseguer. Unwinding and inference control. In *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 1984.

[9] J. Grijpink and J. Prins. New rules for anonymous electronic transactions? An exploration of the private law implications of digital anonymity. In *Digital Anonymity and the Law, Tensions and Dimensions, Information technology and law series*, volume 2, pages 249–269. T.M.C.Asser Press, 2003.

[10] C. Kaufman, R. Perlman, and M. Speciner. *Network Security, PRIVATE Communication in a PUBLIC World*. Prentice Hall, 2nd edition, 2002.

[11] L. Korba, R. Song, and G. Yee. Anonymous Communications for Mobile Agents. In *Proceeding of the 4 thInternational Workshop on Mobile Agents for Telecommunication Applications (MATA'02)a*, volume 2521 of *LNCS*, pages 171–181, 2002.

[12] B. W. Lampson. A note on the Confinement Problem. *Communications of the ACM*, 16(10):613–615, 1973.

[13] R. Leszczyna and J. Górski. Untraceability of mobile agents. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1233–1234, New York, NY, USA, 2005. ACM Press.

[14] M. Luck, P. McBurney, and C. Preist. *Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing)*. AgentLink, 2003.

[15] Z. Maraikar. Resource and service discovery for mobile agent platforms. Master's thesis, Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands, August 2006.

[16] B. Overeinder and F. Brazier. Scalable middleware environment for agent-based internet applications. In *Proceedings of the Workshop on State-of-the-Art in Scientific Computing (PARA'04)*, volume 3732 of *Lecture Notes in Computer Science*, pages 675–679, Copenhagen, Denmark, June 2004. Springer.

[17] A. Pashalidis and C. Mitchell. Limits to Anonymity when Using Credentials. In *Proceedings of the 12th International Workshop on Security Protocols*, LNCS. Springer-Verlag, 2004.

[18] A. Poggi, M. Tomaiuolo, and G. Vitaglione. Security and trust in agent-oriented middleware. In R. Meersman and Z. Tari, editors, *OTM Workshops*, volume 2889 of *Lecture Notes in Computer Science*, pages 989–1003. Springer, 2003.

[19] V. Roth and M. Jalali-Sohi. Concepts and architecture of a security-centric mobile agent server. In *"Proc. of the Fifth International Symposium on Autonomous Decentralized Systems (ISADS 2001)"*, pages 435–442. IEEE Computer Society, 2001.

[20] A. Sabelfeld and A. C. Myers. Language-Based Information-Flow Security. *IEEE Journal on selected areas in communications*, 21(1), 2003.

[21] A. Serjantov. *On the anonymity of anonymity systems*. PhD thesis, University of Cambridge, 2004.

[22] R. van Schouwen. Design and implementation of a secure, decentralized location service for agent platforms. Master's thesis, Department of Computer Sciences, Vrije Universiteit Amsterdam, aug 2006.

[23] M. Warnier, D. de Groot, and F. Brazier. Organized Anonymity in Agent Systems. In *Informal Proceedings of the Fourth European Workshop on Multi-Agent Systems (EUMAS'06)*, 2006.

[24] A. Weimerskirch and G. Thonet. A Distributed Light-Weight Authentication Model for Ad-hoc Networks. In *The 4th International Conference on Information Security and Cryptology (ICISC 2001)*, volume 2288 of *LNCS*, pages 341–354. Springer, 2002.

[25] P. Wurman, M. Wellman, and W. Walsh. The Michigan Internet AuctionBot: a configurable auction server for human and software agents. In *Proceedings of the second international conference on Autonomous agents*, pages 301–308. ACM Press New York, NY, USA, 1998.